# \\CPC-PC\Users\Public\Documents\print\neu.png\\CPC-PC\Users\Public\Documents\print\ccs.png

**New Era University**

**College of Computer Studies**

**Department of Information Technology**

PAYROLL MANAGEMENT SYSTEM



|  |  |
| --- | --- |
| Name                      : | Trajano, Jhon Aaron C. |
| Section                    : | 1IT-1 |
| Subject                    : | PROG2 - IT 121 |
| Date Submitted        : | March 22, 2018 |

**Contents**

[1](#_gjdgxs)

[Views 3](#_1fob9te)

[Login View 3](#_3znysh7)

[Main Path 3](#_2et92p0)

[Alternate Path 3](#_tyjcwt)

[Main View 3](#_3dy6vkm)

[Main Path 3](#_1t3h5sf)

[Alternate Path 6](#_4d34og8)

[Source Code 7](#_2s8eyo1)

[Model Package 7](#_17dp8vu)

[Employee Class 7](#_3rdcrjn)

[EmployeeDA Class 8](#_26in1rg)

[View Package 9](#_lnxbz9)

[LoginView Class 9](#_35nkun2)

[} 10](#_1ksv4uv)

[MainView Class 10](#_44sinio)

[Controller Package 16](#_2jxsxqh)

[AppRunner Class 16](#_z337ya)

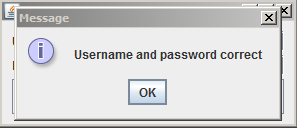
[LoginController 17](#_3j2qqm3)

[MainController 19](#_1y810tw)

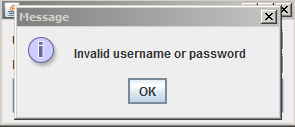
# Views

## Login View

### Main Path



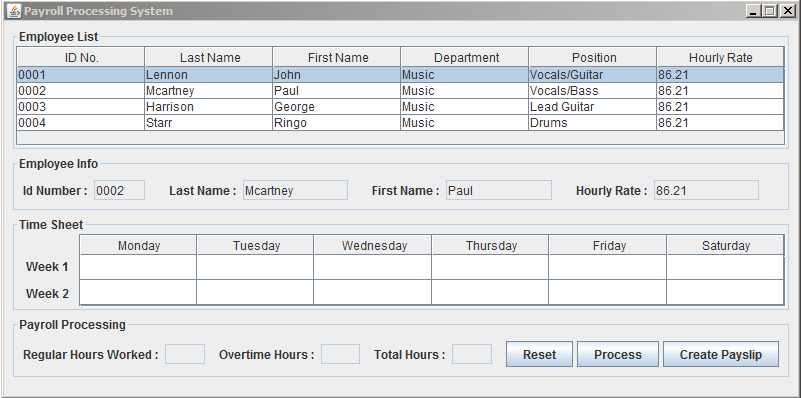
### Alternate Path



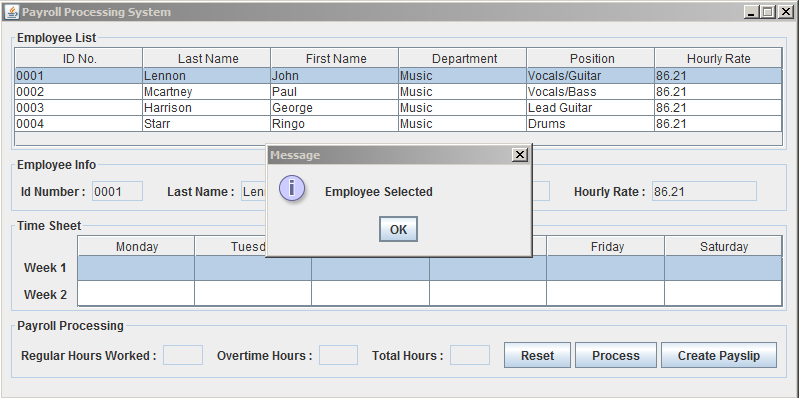
## Main View

### Main Path

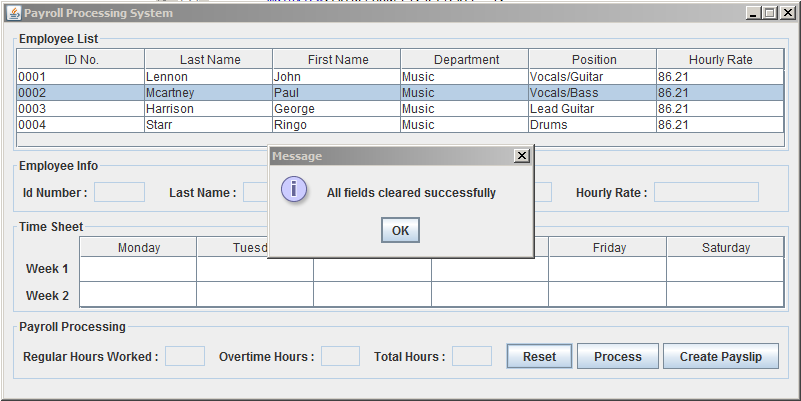
#### Main Frame



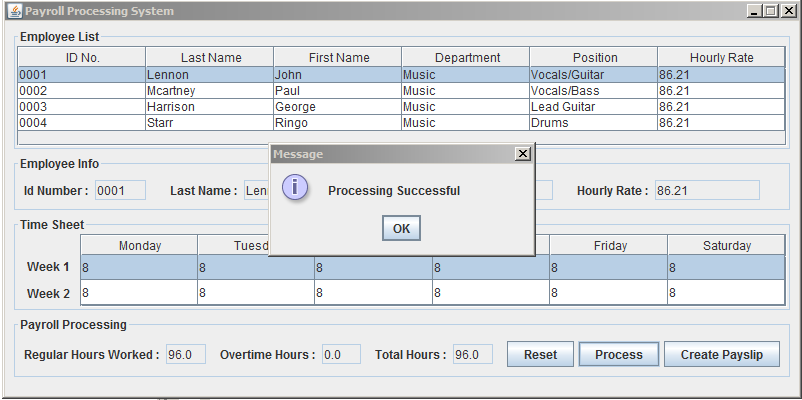
#### Select Employee



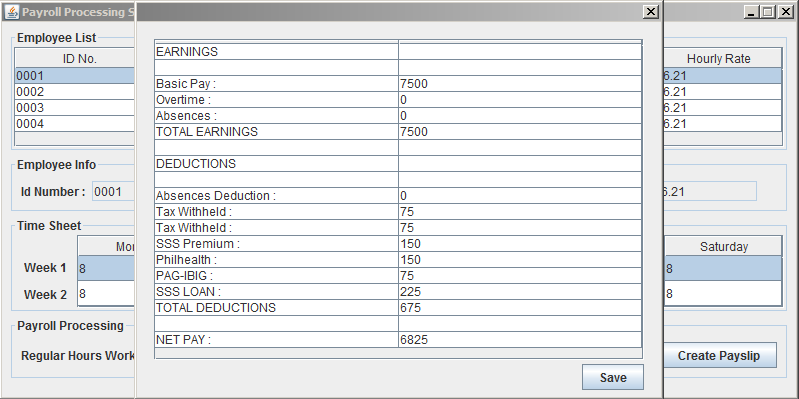
#### Reset Fields



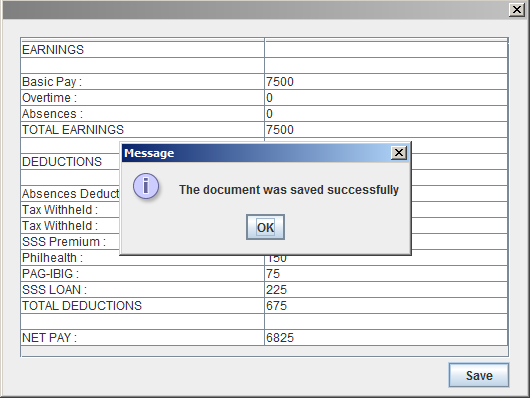
#### Process



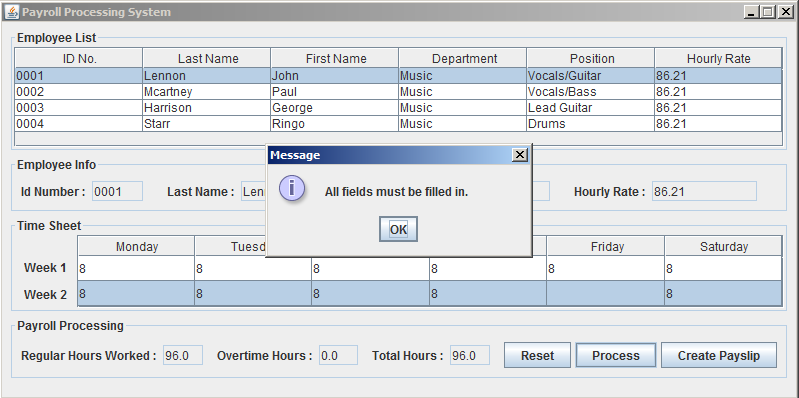
#### Create Pay slip



#### Save



### Alternate Path



# Source Code

## Model Package

### Employee Class

**package** model;

**public** **class** Employee {

**private** String empNo;

**private** String fName;

**private** String lName;

**private** String department;

**private** String position;

**private** Double hourlyRate;

**public** String getEmpNo() {

**return** empNo;

}

**public** **void** setEmpNo(String empNo) {

**this**.empNo = empNo;

}

**public** String getfName() {

**return** fName;

}

**public** **void** setfName(String fName) {

**this**.fName = fName;

}

**public** String getlName() {

**return** lName;

}

**public** **void** setlName(String lName) {

**this**.lName = lName;

}

**public** String getDepartment() {

**return** department;

}

**public** **void** setDepartment(String department) {

**this**.department = department;

}

**public** String getPosition() {

**return** position;

}

**public** **void** setPosition(String position) {

**this**.position = position;

}

**public** Double getHourlyRate() {

**return** hourlyRate;

}

**public** **void** setHourlyRate(Double hourlyRate) {

**this**.hourlyRate = hourlyRate;

}

}

### EmployeeDA Class

package model;

import java.io.FileNotFoundException; import java.io.FileReader; import java.util.ArrayList;

import java.util.List; import java.util.Scanner;

public class EmployeeDA {

private Scanner scan;

private List<Employee> empList;

private String line;

private String [] lineSpecific;

public EmployeeDA() throws FileNotFoundException {

scan = new Scanner(new FileReader("C:\\Users\\kptdaaron\\Documents\\Payslip\\Employee.csv"));

empList = new ArrayList<Employee>();

while(scan.hasNext()) {

Employee employee = new Employee();

line = scan.nextLine();

lineSpecific = line.split(",");

employee.setEmpNo(lineSpecific[0].trim().toString());

employee.setlName(lineSpecific[1].trim());

employee.setfName(lineSpecific[2].trim());

employee.setDepartment(lineSpecific[3].trim());

employee.setPosition(lineSpecific[4].trim());

employee.setHourlyRate(Double.parseDouble(lineSpecific[5].trim().toString()));

empList.add(employee);

}

}

public List<Employee> getEmpList(){

return empList; } }

## View Package

LoginView Class

**package** view;

**import** java.awt.event.ActionListener; **import** java.awt.event.KeyListener; **import** javax.swing.\*;

**import** net.miginfocom.swing.MigLayout;

**public** **class** LoginView **extends** JFrame{

**private** JPanel p;

**private** JLabel u, pw;

**private** JTextField txt;

**private** JPasswordField pwd;

**private** JButton login;

**public** LoginView() {

initializeComponents();

}

**private** **void** initializeComponents() {

setTitle("Login Portal");

setSize(300,130);

setLocationRelativeTo(**null**);

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

p = **new** JPanel();

setContentPane(p);

u = **new** JLabel("Username : ");

txt = **new** JTextField(20);

pw = **new** JLabel("Password : ");

pwd = **new** JPasswordField(20);

login = **new** JButton("Login");

p.setLayout(**new** MigLayout("", "10[][]10", "10[] [] [grow,fill]10"));

p.add(u);

p.add(txt, "pushx, growx, wrap");

p.add(pw);

p.add(pwd, "pushx, growx, wrap");

p.add(login, "span, growx,");

setVisible(**true**);

}

**public** JTextField txt() {

**return** **this**.txt;

}

**public** JPasswordField pwd() {

**return** **this**.pwd;

}

**public** JButton login() {

**return** **this**.login;

}

**public** **void** addActionListener(ActionListener l) {

**this**.login.addActionListener(l);

}

**public** **void** addKeyListener(KeyListener l) {

**this**.txt.addKeyListener(l);

**this**.pwd.addKeyListener(l);

}

### }

### MainView Class

package view;

import javax.swing.\*; import javax.swing.table.DefaultTableModel; import model.Employee;

import model.EmployeeDA; import net.miginfocom.swing.MigLayout; import java.awt.\*;

import java.awt.event.ActionListener; import java.awt.event.MouseListener;

import java.io.FileNotFoundException; import java.io.IOException; import java.text.DateFormat;

import java.text.SimpleDateFormat; import java.util.Calendar; import java.util.GregorianCalendar;

public class MainView extends JFrame{

private JScrollPane empTableWrapper, timeTableWrapper;

public JTable empTable, timetable, psTable;

public JTextField txtRegHour, txtOverHour, txtRegAmount, txtOverAmount, txtNet, txtId, txtLname, txtFname, txtRate, txtTotal;

private JLabel regular, total, over;

float payslip, companyName;

public JButton proc, reset, createPayslip , save;

public String curDate, dateTo;

private EmployeeDA employeeDA;

ListSelectionModel listSelectionModel;

public MainView() throws IOException {

setupUI();

calendar();

timeTable();

employeeTable();

}

public void calendar() {

Calendar calendar = new GregorianCalendar();

DateFormat sdf = new SimpleDateFormat("MM-dd-yyyy");

curDate = sdf.format(calendar.getTime());

calendar.add(Calendar.DAY\_OF\_MONTH, 15);

dateTo = sdf.format(calendar.getTime());

}

private void employeeTable() throws FileNotFoundException {

employeeDA = new EmployeeDA();

DefaultTableModel dtm = new DefaultTableModel()

{

private static final long serialVersionUID = -4762201258155571614L;

public boolean isCellEditable(int row, int column)

{

return false;

}

};

dtm.addColumn("ID No.");

dtm.addColumn("Last Name");

dtm.addColumn("First Name");

dtm.addColumn("Department");

dtm.addColumn("Position");

dtm.addColumn("Hourly Rate");

for(Employee employee:employeeDA.getEmpList())

{

dtm.addRow(new Object[] { employee.getEmpNo(), employee.getlName(), employee.getfName(),

employee.getDepartment(), employee.getPosition(), employee.getHourlyRate()

});

}

empTable.getTableHeader().setReorderingAllowed(false);

empTable.getTableHeader().setResizingAllowed(false);

empTable.setModel(dtm);

}

public void timeTable() {

Object[] columns ={"Monday","Tuesday","Wednesday","Thursday", "Friday", "Saturday"};

Object[][] rows ={{"","","","","",""},

{"","","","","",""}

};

DefaultTableModel dtm = new DefaultTableModel(rows, columns);

timeTable.getTableHeader().setReorderingAllowed(false);

timeTable.setRowHeight(25);

timeTable.setModel(dtm);

}

public void setupUI() throws FileNotFoundException {

setTitle("Payroll Processing System");

setSize(800,400);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setVisible(true);

setLocationRelativeTo(null);

setResizable(false);

JPanel mainPanel = new JPanel();

setContentPane(mainPanel);

mainPanel.setLayout(new MigLayout("", "", "[] [] [] []"));

JPanel empInfoPanel = new JPanel();

empInfoPanel.setLayout(new MigLayout());

empInfoPanel.setBorder(BorderFactory.createTitledBorder("Employee Info"));

JLabel Id = new JLabel("Id Number : ");

txtId = new JTextField(5);

txtId.setEditable(false);

JLabel lName = new JLabel("Last Name : ");

txtLname = new JTextField(10);

txtLname.setEditable(false);

JLabel fName = new JLabel("First Name : ");

txtFname = new JTextField(10);

txtFname.setEditable(false);

JLabel rate = new JLabel("Hourly Rate : ");

txtRate = new JTextField(10);

txtRate.setEditable(false);

empInfoPanel.add(Id);

empInfoPanel.add(txtId, "gapright 20, growx, pushx");

empInfoPanel.add(lName);

empInfoPanel.add(txtLname, "gapright 20, growx, pushx");

empInfoPanel.add(fName);

empInfoPanel.add(txtFname, "gapright 20, growx, pushx");

empInfoPanel.add(rate);

empInfoPanel.add(txtRate, "gapright 20, growx, pushx");

JPanel empPanel = new JPanel();

empPanel.setLayout(new GridLayout(1, 0, 3, 3));

empPanel.setBorder(BorderFactory.createTitledBorder("Employee List"));

empTable = new JTable();

empTableWrapper = new JScrollPane();

empTableWrapper.setPreferredSize(new Dimension(0, 100));

empTableWrapper.setViewportView(empTable);

empPanel.add(empTableWrapper);

JPanel timePanel = new JPanel();

timePanel.setLayout(new MigLayout(""));

timePanel.setBorder(BorderFactory.createTitledBorder("Time Sheet"));

timeTable = new JTable();

timeTableWrapper = new JScrollPane();

timeTableWrapper.setPreferredSize(new Dimension(800,73));

timeTableWrapper.setViewportView(timeTable);

JLabel fWeek = new JLabel("Week 1");

JLabel sWeek = new JLabel("Week 2");

timePanel.add(fWeek,"cell 0 2,gaptop 20, gapleft 10,wrap");

timePanel.add(sWeek,"cell 0 3, gapleft 10");

timePanel.add(timeTableWrapper, "growx, pushx, gapleft 10,east");

JPanel processPanel = new JPanel();

processPanel.setLayout(new MigLayout(""));

processPanel.setBorder(BorderFactory.createTitledBorder("Payroll Processing"));

regular = new JLabel("Regular Hours Worked : ");

processPanel.add(regular);

txtRegHour = new JTextField(5);

txtRegHour.setEditable(false);

processPanel.add(txtRegHour, "gapright 10");

over = new JLabel("Overtime Hours : ");

processPanel.add(over);

txtOverHour = new JTextField(5);

txtOverHour.setEditable(false);

processPanel.add(txtOverHour, "gapright 10");

total = new JLabel("Total Hours : ");

processPanel.add(total);

txtTotal = new JTextField(5);

txtTotal.setEditable(false);

processPanel.add(txtTotal, "gapright 10");

reset = new JButton("Reset");

processPanel.add(reset);

proc = new JButton("Process");

processPanel.add(proc);

createPayslip = new JButton("Create Pay Slip");

processPanel.add(createPayslip);

mainPanel.add(empPanel, "pushx, growx, wrap");

mainPanel.add(empInfoPanel, "pushx, growx, wrap");

mainPanel.add(timePanel,"pushx, growx, wrap");

mainPanel.add(processPanel,"pushx, growx, wrap");

setVisible(true);

}

public JTextField txtRegHour() {

return this.txtRegHour;

}

public JTextField txtOverHour() {

return this.txtOverHour;

}

public JTextField txtRegAmount() {

return this.txtRegAmount;

}

public JTextField txtOverAmount() {

return this.txtOverAmount;

}

public JTextField txtNet() {

return this.txtNet;

}

public JTextField txtId() {

return this.txtId;

}

public JTextField txtLname() {

return this.txtLname;

}

public JTextField txtFname() {

return this.txtFname;

}

public JTextField txtRate() {

return this.txtRate;

}

public JTextField txtTotal() {

return this.txtTotal;

}

public JButton proc() {

return this.proc;

}

public JButton reset() {

return this.reset;

}

public JButton createPayslip() {

return this.createPayslip;

}

public JButton save() {

return this.save;

}

public JTable empTable() {

return this.empTable;

}

public JTable getTimeTable() {

return this.timeTable;

}

public JTable getPayslipTable() {

return this.psTable;

}

public String getCurDate() {

return this.curDate;

}

public String getDateTo() {

return this.dateTo;

}

public void addActionListener(ActionListener l) {

this.proc.addActionListener(l);

this.reset.addActionListener(l);

this.createPayslip.addActionListener(l);

}

public void addMouseListener(MouseListener l) {

this.empTable.addMouseListener(l);

}

}

## Controller Package

### AppRunner Class

**package** controller;

**import** java.io.IOException;

**import** model.Employee;

**import** view.LoginView;

**import** view.MainView;

**public** **class** AppRunner {

**private** **static** MainView *mainView*;

**private** **static** Employee *employee*;

**public** **static** **void** main(String[] args) **throws** IOException {

LoginView loginView = **new** LoginView();

@SuppressWarnings("unused")

LoginController loginController = **new** LoginController(loginView, *mainView*, *employee*);

}

}

### LoginController

package controller;

import java.awt.event.ActionEvent; import java.awt.event.ActionListener;

import java.awt.event.KeyEvent; import java.awt.event.KeyListener;

import java.io.FileNotFoundException; import java.io.IOException;

import javax.swing.JOptionPane;

import model.\*;

import view.\*;

public class LoginController {

private LoginView loginView;

private Employee employee;

public LoginController(LoginView loginView, MainView mainView, Employee employee) {

this.loginView = loginView;

this.loginView.addActionListener(new Action());

this.loginView.addKeyListener(new Action());

}

class Action implements ActionListener, KeyListener {

public void actionPerformed(ActionEvent e) {

isUserValid();

}

@Override

public void keyReleased(KeyEvent e) {

}

@Override

public void keyTyped(KeyEvent e) {

}

@Override

public void keyPressed(KeyEvent e) {

if (e.getKeyCode() == KeyEvent.VK\_ENTER) {

loginView.login().doClick();

}

}

}

public boolean isUserValid() {

String user, pass;

user = loginView.txt().getText();

pass = String.valueOf(loginView.pwd().getPassword());

if(user.equals("admin") && pass.equals("admin")){

JOptionPane.showMessageDialog(null, "Username and password correct");

try {

showMainFrame();

loginView.dispose();

} catch (FileNotFoundException e1) {

e1.printStackTrace();

} catch (IOException e1) {

e1.printStackTrace();

}

}

else{

JOptionPane.showMessageDialog(null, "Invalid username or password");

}

return false;

}

public void showMainFrame() throws IOException

{

MainView mainView = new MainView();

@SuppressWarnings("unused")

MainController controller = new MainController(mainView, employee, loginView);

mainView.setVisible(true);

}

}

### MainController

package controller;

import java.awt.event.ActionEvent; import java.awt.event.ActionListener;

import java.awt.event.MouseEvent; import java.awt.event.MouseListener;

import java.io.FileNotFoundException; import java.io.FileOutputStream;

import java.util.logging.Level; import java.util.logging.Logger;

import javax.swing.\*; import javax.swing.table.DefaultTableModel; import java.awt.\*;

import com.lowagie.text.Chunk; import com.lowagie.text.Document;

import com.lowagie.text.DocumentException; import com.lowagie.text.Paragraph;

import com.lowagie.text.pdf.PdfPTable; import com.lowagie.text.pdf.PdfWriter;

import model.Employee; import net.miginfocom.swing.MigLayout;

import view.LoginView;

import view.MainView;

public class MainController {

private MainView mainView;

public double undertime, totalHours, overtimeHours ,regularHours, tempRate, basicPay, overtimePay, x;

private int absences;

private JButton save;

public MainController(MainView mainView, Employee employee, LoginView loginView) {

this.mainView = mainView;

this.mainView.addActionListener(new Action());

this.mainView.addMouseListener(new Action());

}

class Action implements ActionListener, MouseListener {

@Override

public void actionPerformed(ActionEvent e) {

if(e.getSource().equals(mainView.proc())) {

getTotalHours();

double ut = 0;

if (validCheck()) {

} else {

return;

}

totalHours = getTotalHours();

if(totalHours > 96)

{

overtimeHours = (totalHours - 96);

} else if (overtimeHours < 1) {

overtimeHours = 0;

}

// Compute for overtime

x = absences \* 8;

if (ut > 1) {

undertime = Math.round(112 - totalHours - x);

} else undertime = 0;

regularHours = totalHours - overtimeHours;

basicPay = regularHours \* tempRate;

overtimePay = overtimeHours \* tempRate;

ut = (int) 96 - totalHours;

mainView.txtRegHour().setText(Double.toString(regularHours));

mainView.txtOverHour().setText(Double.toString(overtimeHours));

mainView.txtTotal().setText(Double.toString(totalHours));

JOptionPane.showMessageDialog(null, "Processing Successful");

} else if(e.getSource().equals(mainView.reset())){

reset();

} else if(e.getSource().equals(mainView.createPayslip())){

if(validCheck()) {

payslipDialog();

} else {

JOptionPane.showMessageDialog(null, "All fields must be filled in");

}

}

}

@Override

public void mouseClicked(java.awt.event.MouseEvent evt) {

setToText(evt);

}

@Override

public void mouseEntered(MouseEvent e) {}

@Override

public void mouseExited(MouseEvent e) {}

@Override

public void mousePressed(MouseEvent e) {}

@Override

public void mouseReleased(MouseEvent e) {}

}

public double getTotalHours(){

double totalHours = 0;

if (validCheck()) {

tempRate = Double.parseDouble(mainView.txtRate().getText().toString());

for (int rows = 0; rows < mainView.getTimeTable().getRowCount(); rows ++) {

for (int cols = 0; cols < mainView.getTimeTable().getColumnCount(); cols++) {

totalHours = totalHours+Double.parseDouble(mainView.getTimeTable().getValueAt(rows, cols).toString());

}

}

} else {

JOptionPane.showMessageDialog(null,"All fields must be filled in.");

}

return totalHours;

}

public boolean validCheck() {

if(mainView.getTimeTable().getCellEditor()!= null){

mainView.getTimeTable().getCellEditor().stopCellEditing();

}

else if(mainView.txtId().getText().trim().isEmpty() | mainView.txtLname().getText().trim().isEmpty() |

mainView.txtFname().getText().trim().isEmpty() | mainView.txtRate().getText().trim().isEmpty()) {

return false;

}

for(int i=0;i< mainView.getTimeTable().getRowCount();i++) {

for (int j=0;j<mainView.getTimeTable().getColumnCount();j++) {

String om=mainView.getTimeTable().getValueAt(i,j).toString();

if(om.trim().length()==0) {

return false;

}

}

}

return true;

}

private void reset() {

mainView.txtId().setText("");

mainView.txtLname().setText("");

mainView.txtFname().setText("");

mainView.txtRate().setText("");

mainView.txtRegHour().setText("");

mainView.txtOverHour().setText("");

mainView.txtTotal().setText("");

mainView.timeTable();

JOptionPane.showMessageDialog(null, "All fields cleared successfully");

}

public int countAbsences() {

int occur1 = 0;

int occur2 = 0;

for (int i = 0, rows = mainView.timeTable.getColumnCount(); i < rows; i++) {

if(mainView.timeTable.getValueAt(0, i).equals("0")) {

occur1++;

}

}

for (int i = 0, rows = mainView.timeTable.getColumnCount(); i < rows; i++) {

if(mainView.timeTable.getValueAt(1, i).equals("0")){

occur2++;

}

}

return absences = occur1 + occur2;

}

public double getMonthlyRate() {

double monthlyRate = (tempRate \* 8 \* 261) / 12;

return monthlyRate;

}

public double getAbsentDeduction() {

double absentDeduction = countAbsences() \* tempRate;

return absentDeduction;

}

public void print() throws FileNotFoundException, DocumentException {

try {

Document doc = new Document();

String file = "C:/Users/kptdaaron/Documents/Payslip/" + mainView.getCurDate();

PdfWriter.getInstance(doc, new FileOutputStream(file + "\_" + mainView.txtId().getText() + "\_" + mainView.txtLname().getText() + "\_" + mainView.txtFname().getText() + ".pdf"));

doc.open();

doc.add(new Paragraph("Magical Mystery Tour Foundations, INC"));

doc.add(new Paragraph("PAYSLIP : " + mainView.getCurDate() + " to " + mainView.getDateTo()));

doc.add(new Paragraph("Name : " + mainView.txtLname().getText() + "," + " " + mainView.txtFname().getText()));

doc.add(Chunk.NEWLINE);

PdfPTable pdfTable = new PdfPTable(mainView.getPayslipTable().getColumnCount());

//adding table headers

for (int i = 0; i < mainView.getPayslipTable().getColumnCount(); i++) {

pdfTable.addCell(mainView.getPayslipTable().getColumnName(i));

}

//extracting data from the JTable and inserting it to PdfPTable

for (int rows = 0; rows < mainView.getPayslipTable().getRowCount(); rows++) {

for (int cols = 0; cols < mainView.getPayslipTable().getColumnCount(); cols++) {

pdfTable.addCell(mainView.getPayslipTable().getModel().getValueAt(rows, cols).toString());

}

}

doc.add(pdfTable);

doc.close();

JOptionPane.showMessageDialog(null, "The document was saved successfully");

} catch (DocumentException ex) {

Logger.getLogger(MainView.class.getName()).log(Level.SEVERE, null, ex);

} catch (FileNotFoundException ex) {

Logger.getLogger(MainView.class.getName()).log(Level.SEVERE, null, ex);

}

}

private void payslipDialog() {

JDialog jd = new JDialog();

jd.setLayout(new BorderLayout());

jd.setSize(530,400);

jd.setVisible(true);

jd.setModal(true);

jd.setLocationRelativeTo(null);

jd.setResizable(false);

JPanel p = new JPanel();

p.setLayout(new MigLayout());

mainView.psTable = new JTable();

mainView.psTable.setLayout(new BorderLayout());

JScrollPane jsp = new JScrollPane();

jsp.setViewportView(mainView.psTable);

JLabel earnings = new JLabel("EARNINGS");

save = new JButton("Save");

double te = (getMonthlyRate() / 2) + overtimePay,tax = te \* 0.01, sssp = te \* 0.02, pgbg = te \* 0.01, sssl = te \* 0.03;

double ph = te \* 0.02;

double td = tax+sssp+pgbg+sssl+ph+getAbsentDeduction();

double netPay = te-td;

Object [] column = {"", ""};

Object [][] data = {

{earnings.getText(), ""},

{"", ""},

{"Basic Pay : ", Math.round(getMonthlyRate()/2)},

{"Overtime : ", Math.round(overtimePay)},

{"Absences :", Math.round(countAbsences())},

{"TOTAL EARNINGS", Math.round(te)},

{"", ""},

{"DEDUCTIONS", ""},

{"", ""},

{"Absences Deduction : ", Math.round(getAbsentDeduction())},

{"Tax Withheld : ", Math.round(tax)},

{"Tax Withheld : ", Math.round(tax)},

{"SSS Premium : ", Math.round(sssp)},

{"Philhealth : ", Math.round(ph)},

{"PAG-IBIG :", Math.round(pgbg)},

{"SSS LOAN : ", Math.round(sssl)},

{"TOTAL DEDUCTIONS", Math.round(td)},

{"", ""},

{"NET PAY : " , Math.round(netPay)}

};

DefaultTableModel dtm = new DefaultTableModel(data, column);

mainView.psTable.setModel(dtm);

p.add(jsp, "span, growx, pushx, gap 10 10 10 10");

p.add(save,"right, gapright 10");

jd.add(p, BorderLayout.CENTER);

save.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

mainView.setVisible(false);

try {

print();

} catch (FileNotFoundException | DocumentException e1) {

e1.printStackTrace();

}

mainView.setVisible(true);

}

});

}

public void setToText(java.awt.event.MouseEvent evt) {

DefaultTableModel model = (DefaultTableModel)mainView.empTable().getModel();

int selectedRowIndex = mainView.empTable().getSelectedRow();

mainView.txtId().setText(model.getValueAt(selectedRowIndex, 0).toString());

mainView.txtLname().setText(model.getValueAt(selectedRowIndex, 1).toString());

mainView.txtFname().setText(model.getValueAt(selectedRowIndex, 2).toString());

mainView.txtRate().setText(model.getValueAt(selectedRowIndex, 5).toString());

JOptionPane.showMessageDialog(null, "Employee Selected");

}

}